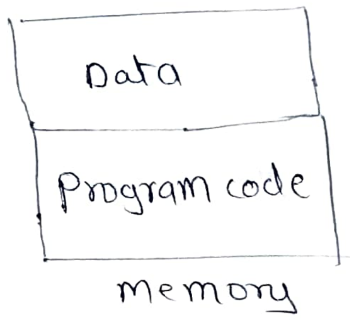
Mastering data structures & Algorithms :-

Topics

Arrays
Matrices
Linked list

} Physical Datastructures
- They define how the data is arranged in memory



Stack
Queues
Trees
Graph
Hashing

} Logical Datastructures
→ This defines how the data can be utilised

- How you keep the data so it can be best utilised by the program so that arrangement of data is data structure.

Recursion
Sortings.

Q why to study data structure?.
→ It is core subject for programmers, In industry if you are working as a programmer you have to use data structures in your applications.

Q At what level should we study DS?
→ Basic → you know what all DS are and how it works.
next level (2) → you know how they work in detail and you are able to do analysis, what are the operations they perform by them and you know how those operations are performed. (Mathematics involved).

Level 3 → you know how to code them, you can develop your own data structures.
(This course cover till level 3)

Q which programming language suitable?.
→ c, c++, Java, c #, python, Javascript, etc

Q Do I have to develop these DS by myself?

→ Some languages have their built in DS, you must know how they work & where to use them.

C is more perfect language for learning DS as they don't have any built'in DS.

- Algorithms are those that are used on these DS. Algorithm is a very vast topic.

ex- face recognition algorithm, vehicle tracking algorithm.

- Study algorithm from abdul Bari youtube channel covered their.

C and C++ concepts :-

① Arrays
② Structure
③ pointers
④ Reference
⑤ parameter passing

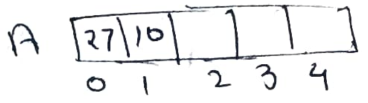
} Related to C

⑥ classes
⑦ constructor
⑧ templates

} Related to C++

Array's Basics :-

```
int A[5];
```



```
A[0] = 27;
A[1] = 10
```

```
int main () {
   int B[5] = { 1, 2, 3, 4, 5 };
```

$\underbrace{\text{int B[5]}}_{\text{Declaration of Array}} = \underbrace{\{1,2,3,4,5\};}_{\text{initialization of Array}}$

```
for (int i = 0; i < 5; i++) {
   printf("%d", B[i]); }
```

→ To print array in c

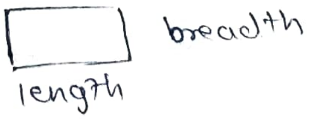
c++ and c print difference

```
int A[10] = { 2, 4, 6, 8, 10, 12 };

// in c++
   for (int x:A) {
      cout << x << endl; }

// in c
   for (int i=0, i< 10, i++) {
      printf("%d\n", A[i]); }
```

structures :-

collection of unsimilar items under one name or grouping unsimilar items.

```
struct Rectangle {
   int length;
   int breadth; }
```



memory consumed will be <u>8 bites</u> and <u>2 ints</u>.

```
int main () {
    struct Rectangle r;          — Declaration
    struct Rectangle r = {10,5};
                            |
         declaration + initializing
```

So r will occupy 8- bites

l	4
b	4

```
    r.length = 15;        -- accessing a member
    r.breadth = 10;
    printf ("Area of Rectangle is %d", r.length * r.breadth);
}
```

<u>Examples of structures</u> —

① complex no. - a+ib (where i - imaginery - $\sqrt{-1}$)

```
        Struct complex {
            int real;
            int img; }
```

② Student -

```
    Struct Student {
        int roll;
        char name [25];
        char dept [10];
        char address [50]; }
```

(you can access them using . operator in main)

This will take 79 bites in memory

4 + 25 + 10 + 50 = 79

```
int main () {
    struct student s;
    s.name = "John";
    s.roll = 10;
    return 0;   }
```

or initialize directly

struct student S = { 10, 'John', ... };

you can also create many students

struct student s [20] = { {10, 'John'..}, {12, 'Tom'..
}, {...}...}; This will declare 20 students.

array of structures.

To print
```
printf ( ~~s~~ "%d", s[0].name );
printf ( "%d", s[1].name );
printf ("%d" , s [0].roll);
```

Practice structures :-

- Declaring variable

```
struct Rectange {
    int  length;
    int  breadth;
} r1, r2, r3; or r1;
```
global

or

outside main
~~Rectangle~~
struct Rectangle R1;
global variable

or

inside main

```
int main () {
    struct Rectangle r1 = {10,5};
    printf ("%d", sizeof (r1));
    return 0; }
```

output - 8 bits as two ints

Note - if you add char x; in struct Rectangle output will be 12 bits but it will use only 1 bits for char bcoz its easy for our machine to read 4 bite's